CS 134: Networks   Problem set 6

Prof. Yaron Singer   Due: **Wednesday 3/22/2017**

1. (30 points)   Consider a variant of the branching process discussed in lecture. The base graph is a rooted, infinite k-ary tree, with $k \geq 2$ an integer (each node has k children and one parent, except for the root which has no parents). We define level n to be the set of nodes in this tree at distance n from the origin.

Let $A_v \in \{0, 1\}$ indicate whether node v is active. If $A_v = 1$, v is active, and if $A_v = 0$, v is inactive. The root node r (the sole node at level $n = 0$) is assumed to be active. For any active node v at level n, the number of its active children at level $n + 1$ is a random variable Z_v (and all sets of active children that have the same size are equally likely).

All the Z_v are mutually independent and distributed identically, according to a probability distribution function P, so that $\mathbb{P}(Z_v = \ell) = P(\ell)$. Assume that $0 < P(0) < 1$ and $0 < P(1) < 1$.

Given these data, the whole (random) tree of active nodes can be constructed inductively. Let X_n be the number of active nodes at level n.

  a. (5 points) What is $\mathbb{E}[X_n]$ in terms of P? (The expectation is taken across all realizations of the random tree.)

  b. (2 points) Let q_n be the probability that $X_n \neq 0$. What is q_0?

  c. (3 points) Is q_n increasing in n, decreasing in n, or neither? Argue from the definition of X_n and the basic properties of the process, without doing any calculations.

  d. (5 points) For $n > 0$, we have $q_n = f(q_{n-1})$. Write out the formula for f and explain your answer. (The function f will involve the values of P.)

  e. (5 points) Recall the function $f : [0, 1] \to [0, 1]$ you described above. Give the sign of its first derivative and the sign of the second derivative. (Here P is again arbitrary, subject to the assumptions given in the statement.)

  f. (5 points) Define $q_\infty = \lim_{n \to \infty} q_n$ when this limit exists. Describe how you can find q_∞ by looking at a plot of some possible q_n vs. n. Give a verbal statement of the meaning of q_∞ (this is required to get full points on this part).

  g. (5 points) Give a necessary and sufficient condition for $q_\infty > 0$ in terms of the values of P only. Explain this condition using a plot of some possible q_n vs. n and your answer in (e).

2. (20 points)   Suppose we have a probability distribution P on the numbers $\{0, 1, \ldots, k\}$ as given in Problem 1. Suppose we have another distribution $\widetilde{P}$ on the same set. We say that $\widetilde{P}$ *first-order stochastically dominates* P if the following condition holds. For every strictly increasing function $r : \{0, \ldots, k\} \to \mathbb{R}$, we have $\sum_{\ell=0}^{k} \widetilde{P}(\ell) r(\ell) > \sum_{\ell=0}^{k} P(\ell) r(\ell)$. In other words, the expectation of $r(Z)$ when Z is drawn from $\widetilde{P}$ always exceeds the expectation of $r(Z)$ when Z is drawn from P. In this sense, $\widetilde{P}$ tends to take higher values.[1]

---

[1] If you're curious, Wikipedia will tell you many different characterizations of first-order stochastic dominance.

a. (5 points) Show that if $\widetilde{P}$ first-order stochastically dominates P and $r : \{0, \ldots, k\} \to \mathbb{R}$ is a strictly *decreasing* function, then $\sum_{\ell=0}^{k} \widetilde{P}(\ell) r(\ell) < \sum_{\ell=0}^{k} P(\ell) r(\ell)$.

b. (7 points) Let f be constructed as in Problem 1(d) based on P, and let $\widetilde{f}$ be constructed in the same way from $\widetilde{P}$. For $q \in (0, 1)$, can you give an inequality relating $\widetilde{f}(q)$ and $f(q)$? State it and justify it. You don't need to give a full proof, but give a clear reason.

c. (8 points) Define $\widetilde{q}_\infty$ as in Problem 1, but now with Z_v having distribution $\widetilde{P}$. When $\widetilde{q}_\infty$ and q_∞ are both positive, give an inequality relating the two and explain it.

3. (15 points) Let P be an $n \times n$ matrix with every entry a nonnegative real number, and the (i, j) entry denoted by p_{ij}. Assume that for every $i \in \{1, \ldots, n\}$ we have $\sum_{j=1}^{n} p_{ij} = 1$. A matrix having these properties is called *row-stochastic*. Assume Q is another row-stochastic matrix.

a. (3 points) Write down a formula for the (i, j) entry of the matrix product $R = QP$.

b. (4 points) For every $i \in \{1, \ldots, n\}$, compute $\sum_{j=1}^{n} r_{ij}$ (give a number) and justify your answer. Note that this is the sum of all entries in row i of R.

c. (4 points) Compute the sum of all the entries in row i of P^2 and P^3, justifying your answer. Compute the row sum of all the entries in row i of P^k, where $k \geq 1$, and justify your answer.

d. (4 points) For each of the three different 5×5 matrices P below, compute $L = P^{100}$ to three decimal digits of precision using your favorite technology. Also compute LP for each case. What do you notice?

$$P_1 = \begin{bmatrix} 0.1 & 0.4 & 0 & 0.3 & 0.2 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0.6 & 0 & 0.2 & 0.1 & 0.1 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.3 & 0.3 & 0.1 & 0.1 & 0.2 \end{bmatrix}, P_2 = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.1 & 0.3 & 0.4 & 0 & 0.2 \\ 0.3 & 0.1 & 0.1 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.3 & 0.2 \\ 0.2 & 0.1 & 0.3 & 0.2 & 0.2 \end{bmatrix}, P_3 = \begin{bmatrix} 0.1 & 0.3 & 0.4 & 0.2 & 0 \\ 0 & 0 & 0.5 & 0.4 & 0.1 \\ 0.2 & 0.2 & 0.3 & 0.2 & 0.1 \\ 0 & 0.3 & 0.5 & 0.1 & 0.1 \\ 0.6 & 0.2 & 0.1 & 0 & 0.1 \end{bmatrix}$$

e. (No credit, just "for fun") A transition matrix is one in which each entry represents the probability of traveling from state i to j in the Markov process—and we'll be covering transition matrices more thoroughly as we move into influence models. View P as the transition matrix of a Markov process on the state space $\{1, 2, \ldots, n\}$, with p_{ij} being the probability of going to state j conditional on being in state i. Give a probabilistic interpretation of P^k, and also probabilistic interpretations of your answer in (d). If you've never encountered Markov chains before in this formalism, try Googling around a bit, reading Wikipedia, watching some instructional videos on the subject...

4. Braess' Paradox (Easley and Kleinberg, 8.4 Q1) (15 points) There are 1,000 cars that must travel from town A to town B. There are two possible routes that each car can take: the upper route through town C or the lower route through town D. Let x be the number of cars traveling on the edge A-C and let y be the number of cars traveling on the edge D-B. The directed graph in Figure 1 indicates that travel time per car on edge A-C is $x/100$ if x cars use edge A-C, and similarly the travel time per car on edge D-B is $y/100$ if y cars use edge D-B. The travel time per car on each of edges C-B and A-D is 12 regardless of the number of cars on these edges. Each driver wants to select a route to minimize his travel time. The drivers make simultaneous choices.

a. **(3 points)** Find Nash equilibrium values of x and y.

b. **(6 points)** Now the government builds a new (one-way) road from town C to town D. The new road adds the path A-C-D-B to the network. This new road from C to D has a travel time of 0 per car regardless of the number of cars that use it. Find a Nash equilibrium for the game played on the new network. What are the equilibrium values of x and y? What happens to total cost of travel (the sum of total travel times for the 1,000 cars) as a result of the availability of the new road?

c. **(6 points)** Suppose now that conditions on edges C-B and A-D are improved so that the travel times on each edge are reduced to 5. The road from C to D that was constructed in part (b) is still available. Find a Nash equilibrium for the game played on the network with the smaller travel times for C-B and A-D. What are the equilibrium values of x and y? What is the total cost of travel? What would happen to the total cost of travel if the government closed the road from C to D?

5. Learning Influence Locally (20 points) Let's say we have a graph $G = (V, E)$ and some opinion data for each node $n \in V$. Assume each node has an opinion that changes over time based on the influence of its neighbors for a particular observed cascade. For a particular observed cascade i, let $opinion_i$ be a vector of timestamped opinions, where $opinion_i[n]$ is a vector such that $opinion_i[n][t]$ is equal to the opinion of node n at time t during observed cascade i, where t ranges from 0 to $\tau - 1$ so that $opinion_i[n]$ has length τ, where opinions are either 0 (node is inactive) or 1 (active).

Now consider one way of modeling how opinions spread across a graph, the Independent Cascade (IC) model. In the IC model, each directed edge (n, m), has some corresponding edge weight $e_{(n,m)} \in (0, 1]$. If n is activated at time step t, then at $t + 1$, n will try activate each of its neighbors $m \in N(n)$ and will succeed with probability $e_{(n,m)}$. Node n will only try to activate its neighbors once, in the time step after n was activated. Moreover, once a node is active, it cannot become inactive.

If we want to fit the independent cascade (IC) model locally to the opinion data, we can use the maximum likelihood estimate (MLE) of the edge weight from n to m, defined as

$$\frac{\sum_i (opinion_i[n][t_{i,n}] * opinion_i[m][t_{i,n} + 1] * (1 - opinion_i[m][t_{i,n}]))}{\sum_i opinion_i[n][t_{i,n}]}$$

where $t_{i,n}$ is the first time n becomes active in cascade i, and we ignore the cascade if n never becomes active. Basically: for a node n, we average across all cascades the likelihood of it (potentially) causing its neighbor m's activation.

The file *network*1.*txt* is a directed graph, where a b corresponds to an edge from a to b. The file *node_opinions.pk* is an array of opinions, where each entry *node_opinions[i]* is a dictionary equivalent to the vectors $opinion_i$ described above, such that $opinion_i[n]$ gives the vector of timestamped opinions described.

(Whew! That was a lot. But don't worry, we'll break it down. It's simpler than you'd think.)

a. **(2 points)** Load the file *opinions.pk*. If you're using python, you can use python's 'pickle' library and the 'load' function to easily turn the file into an array; if you're using another language, we've provided *node_opinions.json*, a JSON-encoded file that you can easily load in any language using the standard JSON encoding and the respective JSON library for your language (though you might have to do some additional string to integer conversion). What is τ

and how many cascades are there? At what timestep (using 0-indexing) does node 13 first have its opinion activated in cascade 6? We will use the convention that if a node is not activated during a cascade then its activation time is τ. (You'll be tempted to do this last part visually by printing the opinion vector, but perhaps do it algorithmically now...)

b. **(2 points)** Now load the file *network*1.*txt*. How many nodes are there? What is the average (out-)degree?

c. **(4 points)** Write a function $doesInfluence(n, m, t, o)$ that takes nodes n and m, time t, and cascade dictionary o, and returns 1 if n is activated at time t and m is activated at time $t + 1$ and not active at time t, or 0 otherwise (or if $t + 1 >= \tau$). For argument $(10, 4, 1, opinion_0)$, what does your function output?

d. **(10 points)** Using the MLE described above, approximate the edge weights of each $e \in E$. What's the estimated weight of edge $(1, 2)$? Of $(26, 21)$? Additionally, submit your estimates as a separate file formatted as a .csv, where entries in the first column correspond to the first node in the edge, entries in the second column correspond to the second node in the edge, and entries in the third column correspond to the edge weight.

e. **(2 points)** What node has the highest average edge weight (for outgoing edges)? What node has the lowest average edge weight for outgoing edges? What node, on average, is activated first (in cascades where it activates)? What node, on average, is activated last?
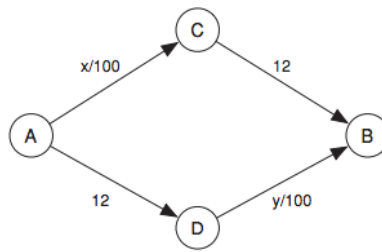
Figure 1: A traffic network.